



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Am

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/916,460	07/26/2001	Gowri Rajaram	UTL 00103	8739
32968	7590	06/17/2005	EXAMINER	
SHRADER, LAWRENCE J				
ART UNIT		PAPER NUMBER		
2193				

DATE MAILED: 06/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.	RAJARAM ET AL.	
09/916,460		
Examiner Lawrence Shrader	Art Unit 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 06 December 2004.
2a) This action is FINAL. 2b) This action is non-final.
3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-58 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) Claim(s) _____ is/are allowed.
6) Claim(s) 1-58 is/are rejected.
7) Claim(s) _____ is/are objected to.
8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 12/15/2004.
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
5) Notice of Informal Patent Application (PTO-152)
6) Other: _____.

DETAILED ACTION

1. Receipt of the Information Disclosure Statement filed on 12/15/2004 is acknowledged and it has been considered.
2. Claims 1 – 58 remain rejected.

Specification

3. The objection to the abstract of the disclosure because its length exceeds 150 words is withdrawn in view of the amendment filed on 12/06/2004.

Double Patenting

4. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claim 1 is provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claim 1 of copending Application No. 09/916,900. Although the conflicting claims are not identical, they are not patentably distinct

from each other because all the limitations of claim 1 are found in the limitations of claims 1 – 58 in the 09/916,900. The concept of sizing code sections is taught in both applications.

09/916,460

09/916,900

<i>"storing wireless device system software in a plurality of current code sections; (claim1)</i>	<i>"storing system software for the wireless device in a plurality of current code sections; (claim 1)</i>
<i>receiving a new code section; (claim 1)</i>	<i>receiving a new code section;" (claim 1)</i>
<i>resizing current code sections; (claim 1)</i>	<i>"...sizing code sections to accommodate arranged symbol libraries..." (claim 28)</i>
<i>arranging the new code section with the current code sections to form updated system software for the wireless device; (claim1)</i>	<i>"arranging the new code section with current code sections to form updated system software for the wireless device; (claim 1)</i>
<i>executing the updated system software."</i> <i>(claim 1)</i>	<i>executing the updated system software."</i> <i>(claim 1)</i>

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1 – 58 are rejected under 35 U.S.C. 103(a) as being unpatentable over Henerlau et al., U.S. Patent 6,442,660 (hereinafter referred to as Henerlau) in view of art of record Hansson, U.S. Patent 6,023,620.

In regard to claim 1:

In a wireless communications device, a method for reorganizing software instructions stored in a memory, the method comprising:

"storing wireless device system software in a plurality of current code sections;

receiving a new code section;

resizing current code sections;

arranging the new code section with the current code sections to form updated system software for the wireless device;

executing the updated system software."

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application (Figures 2 and 3; column 2, lines 1 – 15 and column 12, lines 20 – 28; column 8, lines 47 – 55), but does not disclose a wireless system. However, Hansson discloses a wireless device that receives, reorganizes, and stores system software in a plurality of code sections, arranges the code, and executes (See Figure 1; column 4, lines 16 – 54). Therefore, it would have been obvious to one skilled in the art to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by

Henerlau, because the combination allows code section update to a remote system as taught by Hansson at column 1, lines 6 – 10.

In regard to claim 2, incorporating the rejection of claim 1:

"...further comprising:

identifying a current code section for updating;

wherein arranging the new code section with the current code sections to form updated system software includes replacing the identified current code section with the new code section."

Henerlau discloses identifying a current code section for updating (column 4, lines 1 – 39).

In regard to claim 3, incorporating the rejection of claim 2:

"...further comprising:

forming the system software into a first plurality of symbol libraries, each symbol library comprising at least one symbol;

arranging the first plurality of symbol libraries into a second plurality of code sections."

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application, but neither Henerlau nor Hansson discloses symbol libraries. However, Li discloses symbol libraries for analyzing class relationships. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by Henerlau, and further combined with the symbol library as taught by Li because the

combination allows the code to be parsed and reconfigured as taught by Li at column 5, lines 15 – 23.

In regard to claim 4, incorporating the rejection of claim 3:

“...wherein receiving a new code section includes receiving the new code section via a wireless communications device air interface.”

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application, but does not disclose a wireless system interface. However, Hansson discloses reception of new code via a wireless device (See Figure 1). Therefore, it would have been obvious to one skilled in the art to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by Henerlau, because the combination allows code section update to a remote system as taught by Hansson at column 1, lines 6 – 10.

In regard to claim 5, incorporating the rejection of claim 4:

“...wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes starting symbol libraries at the start of code sections;

wherein storing wireless device system software in a plurality of current code sections includes storing the start of code sections at corresponding start addresses;

the method further comprising:

maintaining a code section address table cross-referencing code section identifiers with corresponding start addresses.”

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39).

In regard to claim 6, incorporating the rejection of claim 5:

“...wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes arranging symbols to be offset from their respective code section start addresses;

the method further comprising:

maintaining a symbol offset address table cross-referencing symbol identifiers with corresponding offset addresses, and corresponding code section identifiers.”

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39).

In regard to claim 7, incorporating the rejection of claim 6:

“...wherein storing the start of code sections at corresponding start addresses includes:

creating a second plurality of contiguously addressed memory blocks;

identifying each memory block with a corresponding code section;

storing code sections in the identified memory blocks.”

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 10 – 46).

In regard to claim 8, incorporating the rejection of claim 7:

“...wherein arranging the first plurality of symbol libraries into a second plurality of code sections includes sizing the code sections to accommodate arranged symbol libraries;

wherein creating a second plurality of continuously addressed memory blocks includes sizing memory blocks to accommodate corresponding code sections.”

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application, but neither Henerlau nor Hansson discloses symbol libraries. However,

Li discloses symbol libraries for analyzing class relationships. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by Henerlau, and further combined with the symbol library as taught by Li because the combination allows the code to be parsed and reconfigured as taught by Li at column 5, lines 15 – 23.

In regard to claim 9, incorporating the rejection of claim 8:

“...wherein sizing memory blocks to accommodate corresponding code sections includes sizing the code sections to accommodate sizes larger than the arranged symbol libraries.”

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application, but neither Henerlau nor Hansson discloses symbol libraries. However, Li discloses symbol libraries for analyzing class relationships. Therefore, it would have been obvious to one skilled in the art at the time the invention was made to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by Henerlau, and further combined with the symbol library as taught by Li because the combination allows the code to be parsed and reconfigured as taught by Li at column 5, lines 15 – 23.

In regard to claim 10, incorporating the rejection of claim 9:

“...wherein resizing current code sections includes resizing the memory blocks in which corresponding resized code sections are stored.”

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application (Figures 2 and 3; column 2, lines 1 – 15 and column 12, lines 20 – 28).

In regard to claim 11, incorporating the rejection of claim 10:

"...wherein receiving a new code section includes receiving a new code section having a first size;

wherein identifying a current code section for updating includes identifying a current code section having a second size, less than the first size;

wherein resizing the memory blocks in which corresponding resized code sections are stored includes:

increasing the size of a memory block associated with the identified current code section to at least the first size;

replacing the identified current code section, stored in the corresponding memory block, with the new code section."

Henerlau discloses receiving, resizing, and executing a plurality of code sections in an embedded application (Figures 2 and 3; column 2, lines 1 – 15 and column 12, lines 20 – 28).

In regard to claim 12, incorporating the rejection of claim 11:

"...wherein resizing the memory blocks in which corresponding resized code sections are stored includes, in response to measuring the size of symbol libraries arranged within the corresponding code sections, resizing code sections to more closely match the symbol library sizes arranged within."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 13, incorporating the rejection of claim 12:

"...wherein resizing code sections to more closely match the symbol library sizes arranged within includes optimally resizing code sections to further subsequent code section resizing and updating operations."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 14, incorporating the rejection of claim 12:

"...further comprising:

using the start addresses from code section address table, measuring the current code sections sizes;

using the symbol offset addresses from the symbol offset address table, measuring the size of the symbol libraries arranged within corresponding code sections."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 15, incorporating the rejection of claim 14:

"...wherein forming the system software into a first plurality of symbol libraries includes forming end symbols to signify the end of symbol libraries;

wherein measuring the size of the symbol libraries arranged within corresponding code sections includes using the offset addresses of the end symbols to measure the size of symbol libraries."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 16, incorporating the rejection of claim 14:

"...wherein forming the system software into a first plurality of symbol libraries includes forming size symbols to signify the size of symbol libraries;

wherein measuring the size of the symbol libraries arranged within corresponding code sections includes accessing the size symbols to measure the size of symbol libraries."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 17, incorporating the rejection of claim 15:

"...further comprising:

following the resizing of the code sections, changing the code section start addresses."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 18, incorporating the rejection of claim 16:

"...further comprising:

measuring the size of the new code section;

"...in response to measuring the size of symbol libraries arranged within corresponding code sections, and measuring the size of the new code section, determining if the new code section can be arranged with the current code sections."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 19, incorporating the rejection of claim 18:

"...wherein determining if the new code section can be arranged with the current code section includes determining the size of unused memory blocks;

wherein arranging the new code section with the current code sections to form updated system software for the wireless device includes storing the new code section in the

unused memory block, if the size of the unused memory block is greater than, or equal to the new code section size."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 20, incorporating the rejection of claim 18:

"...further comprising:

calculating the code section sizes;

in response to calculating the code section sizes, generating a compaction schedule;

temporarily moving code sections into a file system section;

wherein arranging the new code section with the current code sections to form updated system software for the wireless device includes storing the code sections from the file system section into the memory blocks to maintain contiguous addressing, in response to the compaction schedule."

Hernerlau discloses calculating code section sizes, temporarily storing differences in a relocation table, and storing the embedded application according to a compaction schedule as found in Figure 3 (See also the Abstract and column 3, lines 35 – 43).

In regard to claim 21, incorporating the rejection of claim 20:

"...further comprising:

in response to changing the start addresses of code sections stored within the memory blocks, generating an updated code section address table;

wherein executing the updated system software includes using the updated code section address table after arranging the new code section with the current code sections."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 22, incorporating the rejection of claim 21:

“...wherein receiving a new code section includes receiving a new code section with an updated symbol offset address table;

wherein executing the updated system software includes using the updated symbol offset address table after arranging the new code section with the current code sections.”

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 32 – 39; Figures 1 and 2).

In regard to claim 23, incorporating the rejection of claim 11:

“...wherein receiving a new code section includes receiving a compaction instruction set including code section resizing instructions and a compaction schedule;

wherein resizing current code sections includes resizing code sections in response to the code section resizing instructions.”

Henerlau discloses calculating code section sizes, temporarily storing differences in a relocation table, and storing the embedded application according to a compaction schedule as found in Figure 3 (See also the Abstract and column 3, lines 35 – 43).

In regard to claim 24, incorporating the rejection of claim 23:

“...further comprising:

temporarily moving code sections into a file system section;

wherein arranging the new code section with the current code sections to form updated system software for the wireless device includes storing the code sections from the file

system section into memory blocks to maintain contiguous addressing, in response to the compaction schedule."

Henerlau discloses calculating code section sizes, temporarily storing differences in a relocation table, and contiguously storing the embedded application according to a compaction schedule as found in Figure 3 (See also the Abstract and column 3, lines 35 – 43).

In regard to claim 25, incorporating the rejection of claim 24:

"...wherein receiving a new code section includes receiving a new code section with an updated code section address table and an updated symbol offset address table;

wherein executing the updated system software includes using the updated code section address table and updated symbol offset address table after arranging the new code section with the current code sections."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 22 – 46; Figures 1 and 2).

In regard to claim 26, incorporating the rejection of claim 6:

"...further comprising:

loading the code section address table and symbol offset address table into a volatile memory;

in response to loading the code section address table and symbol offset address table into the volatile memory, executing system software;

resetting the wireless communications device;

in response to resetting, loading the updated code section address table and the updated symbol offset address table into volatile memory;

wherein executing the updated system software includes executing the updated system software in response to loading the updated code section address table and updated symbol offset address table into memory."

Henerlau discloses building a new version of the code using a relocation table as a cross reference establishing the proper address (column 4, lines 22 – 46; Figures 1 and 2), but does not disclose a wireless system. However, Hansson discloses a wireless device that receives, reorganizes, and stores system software in a plurality of code sections, arranges the code, and executes (See Figure 1; column 4, lines 16 – 54). Therefore, it would have been obvious to one skilled in the art to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by Henerlau, because the combination allows code section update to a remote system as taught by Hansson at column 1, lines 6 – 10.

In regard to claim 27, incorporating the rejection of claim 26:

"...wherein resizing current code sections includes suspending the operation of the system software."

Henerlau discloses resizing includes suspension of the software operation (column 4, lines 22 – 25).

In regard to claim 28:

In a wireless communications device, a method for reorganizing software instructions stored in a memory, the method comprising:

storing wireless device system software in a plurality of current code sections with the start of code sections at corresponding start addresses by creating a second plurality of contiguously addressed memory blocks, identifying each memory block with a corresponding code section, and storing code sections in identified memory blocks;

receiving a new code section via a wireless communications device air interface;
identifying a current code section for updating;
calculating the code section sizes;
in response to calculating the code section sizes, generating a compaction schedule;
resizing current code sections;
following the resizing of the current code sections, changing the code section start addresses;
temporarily moving code sections into a file system section;
replacing the identified current code section with the new code section by storing the code sections from the file system section into memory blocks to maintain contiguous addressing, in response to the compaction schedule;
executing the updated system software."

Hernerlau discloses calculating code section sizes, temporarily storing differences in a relocation table, and storing the embedded application according to a compaction schedule as found in Figure 3 (See also the Abstract and column 3, lines 35 – 43; column 8, lines 47 – 55). Henerlau also discloses receiving, resizing, and executing a plurality of code sections in an embedded application (Figures 2 and 3; column 2, lines 1 – 15 and column 12, lines 20 – 28; column 8, lines 47 – 55), but does not disclose a wireless system. However, Hansson discloses a wireless device that receives, reorganizes, and stores system software in a plurality of code sections, arranges the code, and executes (See Figure 1; column 4, lines 16 – 54). Therefore, it would have been obvious to one skilled in the art to combine the software update system in a wireless device as taught by Hansson with the resizing of the code sections as taught by

Henerlau, because the combination allows code section update to a remote system as taught by Hansson at column 1, lines 6 – 10.

In regard to claim 29 (a system), rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 1.

In regard to claim 30 (a system), incorporating the rejection of claim 29, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 2.

In regard to claim 31 (a system), incorporating the rejection of claim 30, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 3.

In regard to claim 32 (a system), incorporating the rejection of claim 31, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 4.

In regard to claim 33 (a system), incorporating the rejection of claim 32, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 5.

In regard to claim 34 (a system), incorporating the rejection of claim 33, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 6.

In regard to claim 35 (a system), incorporating the rejection of claim 34, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 7.

In regard to claim 36 (a system), incorporating the rejection of claim 35, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 8.

In regard to claim 37 (a system), incorporating the rejection of claim 36, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 9.

In regard to claim 38 (a system), incorporating the rejection of claim 37, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 10.

In regard to claim 39 (a system), incorporating the rejection of claim 38, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 11.

In regard to claim 40 (a system), incorporating the rejection of claim 39, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 12.

In regard to claim 41 (a system), incorporating the rejection of claim 40, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 13.

In regard to claim 42 (a system), incorporating the rejection of claim 40, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 14.

In regard to claim 43 (a system), incorporating the rejection of claim 42, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 15.

In regard to claim 44 (a system), incorporating the rejection of claim 42, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 16.

In regard to claim 45 (a system), incorporating the rejection of claim 43, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 17.

In regard to claim 46 (a system), incorporating the rejection of claim 45, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 18.

In regard to claim 47 (a system), incorporating the rejection of claim 45, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 18.

In regard to claim 48 (a system), incorporating the rejection of claim 45, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 19.

In regard to claim 49 (a system), incorporating the rejection of claim 45, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 20.

In regard to claim 50 (a system), incorporating the rejection of claim 49, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 21.

In regard to claim 51 (a system), incorporating the rejection of claim 50, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 22.

In regard to claim 52 (a system), incorporating the rejection of claim 39, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 23.

In regard to claim 53 (a system), incorporating the rejection of claim 52, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 24.

In regard to claim 54 (a system), incorporating the rejection of claim 53, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 25.

In regard to claim 55, incorporating the rejection of claim 29:

“...wherein the code storage section and file system section memories are nonvolatile memories.”

Henerlau discloses a ROM system for storage of the code (See Figure 1).

In regard to claim 56 (a system), incorporating the rejection of claim 34, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 26.

In regard to claim 57 (a system), incorporating the rejection of claim 56, rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 27.

In regard to claim 58 (a system), rejected for the same corresponding reasons put forth in the rejection of the corresponding method of claim 28.

Response to Arguments

Applicant's arguments filed on 12/06/2004 have been fully considered but they are not persuasive:

The Applicant has argued in regard to claims 1, 29, and 58 that Henerlau does not disclose resizing of code sections. However, at column 8, lines 47 – 55 Henerlau clearly discloses resizing the code section by adjusting the heap pointer, therefore, the well known concept of resizing or compacting code sections in order to more efficiently use memory is clearly taught.

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Lawrence Shrader whose telephone number is (703) 305-8046. The examiner can normally be reached on M-F 08:00-16:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Lawrence Shrader
Examiner
Art Unit 2193

6 June 2005

Lawrence Shrader
KAKALI CHAKI
SUPERVISOR PATENT EXAMINER
TECHNOLOGY CENTER 2100